



LINUX Expert

If your website expects visitors to send sensitive data, then you will need to apply some sensible security measures. Jon Thompson explains how you can encrypt information using SSL



OPEN SOURCE WEATHER

It probably hasn't escaped your attention that the BBC has a new-look weather service. What's less widely known is that the underlying software runs on Linux. Weatherscape XT, from New Zealand-based Metra (www.metra.info), uses OpenGL to generate its virtual fly-bys of the nation, with data supplied from a back-end MySQL database. It also provides full cross-platform support to versions running on OS X and Windows.

Now in use with all the BBC's visual broadcasting services, Weatherscape XT is thought to be the highest-profile Linux application deployment to date. The system replicates data to remote studios around the country, too, so the latest weather information is always at hand for regional forecasts.

MOBILE NOKIA DEVELOPMENTS

Nokia has announced the new model 770 Internet Tablet and accompanying open source-based Maemo Development Platform. The device uses a Texas Instruments OMAP processor and a dedicated digital signal processor. The 770 measures 14cm x 8cm and has an impressively high-resolution 800x480 screen, capable of displaying up to 65,536 colours. Under the hood is a Linux 2.6 kernel, an X11 server and the GTK+ toolkit, along with the new Hildon graphical user interface.

The Maemo website says it contains "popular open source software components, which are widely deployed in the leading Linux desktop distributions today," and includes tools specifically developed to ease the migration of existing code to non-i386 platforms.

CUBAN WINDOWS CRISIS

Roberto del Puerto, Cuba's Director of the State Office of Information Technology, has announced that the world's only Caribbean communist country plans to migrate all state computers from Microsoft Windows to Linux. Speaking to state-controlled Cuban daily newspaper *Juventud Rebelde*, he said that around 1,500 computers already ran Linux. Del Puerto said his office was working on a legal framework to enable the migration to take place.

In the beginning, the internet was insecure and had no built-in encryption. When the web was created, it was assumed that the Hypertext Transfer Protocol (HTTP) would sit on top of any of a number of secure data transport mechanisms. Sensibly, this modular approach allowed sites to use different encryption schemes as technology advanced. Nearly 15 years on, however, the default is still to transmit HTTP as insecure, clear text.

Websites that collect sensitive information need to encrypt transmissions between the client and server so details such as credit card numbers cannot be intercepted by criminals. This has to work seamlessly and provide the best possible security. In 1994, Netscape came up with the Secure Sockets Layer (SSL). The web still uses SSL today; here we'll look at how you can add it to your own site.

There are different ways to use SSL on Linux. Apache-SSL (www.apache-ssl.org) is a version of the Apache web server built using a version of SSL called OpenSSL (www.openssl.org). Apache has an open source SSL module, which we'll use to secure connections to the web server that we built in *Linux Expert*, *Computer Shopper* July 2005. You can find a PDF of that article on this month's cover disc.

SSL sits between applications such as Apache and the server's networking software. It encrypts traffic before lower-level software bundles it up into packets ready for transmission. It also decrypts incoming data. SSL doesn't just secure web traffic (HTTP) though. It can also protect other text-based application protocols, including POP3 and SMTP.

KEYS TO THE KINGDOM

SSL uses a mixture of encryption and trusted certificates to ensure that information exchanged between your site and other people is encrypted, and that users can be sure that it's your site and not that of an impostor.

There are two main encryption techniques: symmetric and asymmetric. Symmetric encryption is so named because it uses the same key to encrypt and decrypt messages. But if the key should fall into the wrong hands, someone else could decrypt and read the protected information. So symmetric encryption is safe to use only if both sender and receiver already know the key or can send it over a secure channel.

Asymmetric encryption, which is used by SSL, is a safer method of securing a message. In asymmetric encryption, you use a publicly known key to encrypt messages, but need a secret, private key to decrypt them again. A random process usually generates the private key first, and clever maths derives the public key from this.

As an example, suppose Bob wants to send a secure message to Alice. He uses Alice's public key as one half of the input to an asymmetric encryption algorithm. The other input is the

message itself. Bob sends the resultant gibberish to Alice, safe in the knowledge that the public key can play no part in its decryption.

Alice feeds her private key into her copy of the algorithm along with the encrypted message. The output is the original text. No other key will decrypt the message. This means that anyone can send Alice a secret message, but only she can decode it.

The problem is that Jim also has access to Alice's public key, so he might decide to pretend to be Bob. The message he sends, while secure, would contain fake information. How would Alice know that the message hadn't come from Bob? One way is to have a trusted third party that will vouch for Bob in a way that it won't for Jim.

SSL does this using a system of public electronic certificates issued by trusted, independent organisations known as certificate authorities (CAs). When a web server receives a connection request, it sends a copy of its certificate to the web browser making the request. The browser checks a website's certificate against a list of CAs. If a trusted source didn't issue it, you'll see a warning message. If it did, the client automatically uses the public key contained in the certificate to encrypt a random symmetric key, which it sends to the server.

The server decrypts this package to discover the client's symmetric key. It uses this to encrypt its own symmetric key, which it then sends back to the client, so only the client and server know each other's symmetric key. They can now encode and decode the data that will pass between them at top speed. Symmetric encryption is much faster than asymmetric encryption. See our diagram opposite for an overview of the conversation between client and server.

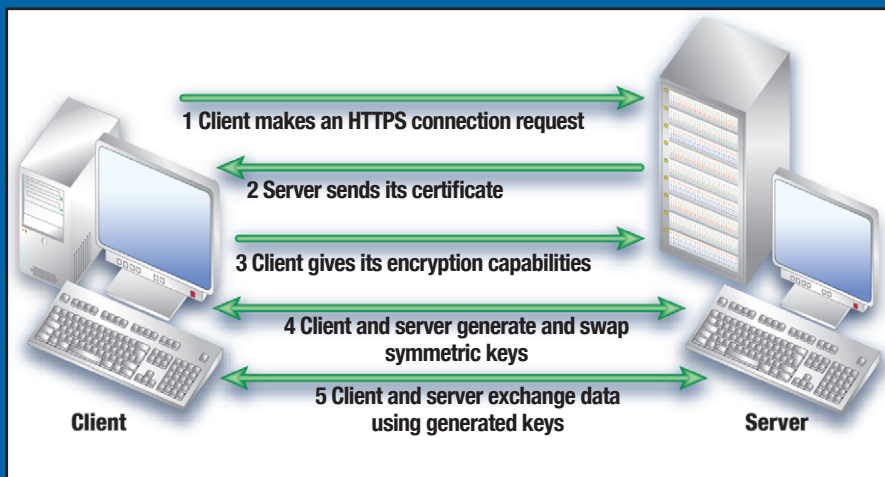
By default SSL-secured HTTP (HTTPS) data uses port 443 rather than port 80, which is usually reserved for regular HTTP web traffic. To the user, all that seems to happen is that a small padlock or similar icon appears somewhere in the browser and the HTTP part of the URL in the browser address bar changes to HTTPS. Mozilla Firefox goes one better and turns the URL field in the browser a nice shade of beige.

It's worth sounding a word of caution before we continue. SSL, in all its implementations, from proprietary code to open source, only secures the communications channel established between client and server. The data is safe as it travels across the internet, but it's not encrypted on the client or server.

At the server end, it hands the decrypted data over to whichever software handles it next. So using a PHP or a Perl script to collect credit card details to store in a MySQL database does not automatically store that data securely. Over the years, many sites have assumed that using SSL makes them secure



HOW THE SSL CONNECTION WORKS



The process of establishing an SSL connection, known as the handshake phase, requires several items of information to pass between the client and the server. The first is the server's certificate, which contains its public key. The client, after checking and accepting the certificate, uses the public key to encrypt its response, which is to tell the server of its encryption capabilities and to send it a newly generated symmetric key. This is an encrypted message, and only the server can decrypt it to retrieve the client's symmetric key.

The server then generates its own symmetric key for the client to use, and encrypts it using the client's symmetric key, before sending it back. The client then decrypts the server's symmetric key. Now both parties have swapped keys securely, they start the data exchange phase of the connection, safe in the knowledge that only they know the symmetric keys. These are then discarded as new keys are generated for each future connection.

but found that once a hacker gains access to the server, he can identify any information he likes.

START YOUR SERVERS

If you're using an up-to-date Linux distribution that ships with Apache 2 you'll find the open-source SSL module, called `mod_ssl`, is already bundled with the server and just needs to be enabled in `httpd.conf`. This is the simplest option, and we recommend it. If you have already installed Apache as per our July article, skip down to where we explain how to enable the SSL module.

If you don't have Apache installed and don't have a graphical package installation tool, copy the Apache2 RPM from your distribution or from the Apache website (www.apache.org). Then install it using the command:

```
# rpm -i <package>#
```

Here `<package>` is the name of your Apache2 RPM.

If you choose to use Apache 1.3.x, you'll have to install it by hand. For that, you'll need the Apache source code. This is also available from the Apache site. Assuming you've placed the downloaded tarball in `/usr/local/src`, you can unpack it as follows:

```
# gzip -d <apache_nnn>.tar.gz#
# tar xvf <apache_nnn>.tar#
```

Here `<apache_nnn>` is the latest release of version 1.3. You'll also need `mod_ssl`. If it is not on your distribution CD, you can obtain and install it. Just visit www.modssl.org and find out the name of the latest release. At the time of writing it was version 2.8.22-1.3.33. The references to `mod_ssl.xxx.tar.gz` below refer to the file containing the latest version of `mod_ssl`, which will be downloaded from www.modssl.org. Currently these references would read `mod_ssl.2.8.22-1.3.33.tar.gz`.

```
# cd /usr/local/src#
# wget -q http://www.modssl.org/
```

```
source/mod_ssl.xxx.tar.gz#
# tar zxvf mod_ssl.xxx.tar.gz#
# cd mod_ssl.xxx.tar.gz#
# ./configure -with-apache=../apache_1.3.x#
```

Here `../apache_1.3.x` is the source directory for the Apache server. Because Apache 1.3 cannot directly support SSL, the `mod_ssl` installation process modifies the Apache source code. Because of this, we must rebuild the Apache server. In the Apache source directory, type:

```
# ./configure -prefix=/usr/local/apache
-enable-module=ssl#
# make#
# make install#
```

Apache2 is a better option for several reasons, including security and ease of installation. We'll base the rest of the configuration on this release.

Once configured to use `mod_ssl`, Apache will refuse to start up until we generate a private key to decode connection requests and obtain a certificate proving our server's identity. To test our installation, we must generate our own test certificate. After that, we'll show you how to request a real one from a CA.

First, we use the multipurpose `openssl` command to generate a 1,024-bit private key using the RSA encryption algorithm, which we'll store in `/usr/local/apache/conf/ssl-key`:

```
# openssl genrsa -out /etc/apache2/ssl.key/server.key 1024#
Generating RSA private key, 1024 bit long modulus
e is 65537 (0x10001)
```

The server's private key is stored in the file called `server.key`, the contents of which look like a jumble of letters and numbers. You might be able to decipher the jumble with the `openssl` command:

```
# openssl rsa -noout -text -in /etc/apache2/ssl.key/server.key#
```

This produces meaningless hexadecimal, but it does tell you what the different sections of the file mean. You can also show the server's public key on the screen using the `openssl` command:

```
# openssl rsa -in /etc/apache2/ssl.key/server.key -pubout#
```

Now we have our private key we need to request a signed certificate from a recognised CA. To do this, we have to create a certificate signing request (CSR). Again, the `openssl` command will generate this for us. It contains encrypted information that identifies our site and its public key, which it derives from the private one.

We need various pieces of information at hand for this, so it's best to sit down and write it all out first:

Country code	A 2-digit code (GB, US, etc)
State/Province	A piece of text identifying your county, etc
Locality	Where the server is sited
Organisation name	Your company or enterprise name
Organisational unit	Whichever unit within the company owns the server
Common name	Usually your server's hostname
Email	A contact address (the system manager or web master)
Challenge password	Optional; leave this blank
Optional company name	Optional; leave this blank

To enter the details and create the CSR, type:

```
# openssl req -new -key /etc/apache2/ssl.key/server.key -out /etc/apache2/ssl.csr/server.csr#
```

This begins generating the CSR into the file called `server.csr` in the `/etc/apache2/ssl/csr/` directory. You can leave any fields blank by entering a full stop. If you just press return, `openssl` will assume you want to use the default values shown in the square brackets.

Now we have our CSR, we can test our work so far by generating our own certificate:

```
# openssl x509 -req -days 365 -in /etc/apache2/ssl.csr/server.csr
-signkey /etc/apache2/ssl.key/server.key -out /etc/apache2/ssl.crt/server.crt#
Signature ok
subject=/C=GB/ST=London/L=London/O=CSshopper/OU=Web Server /CN=Jon Thompson/emailAddress=jon@csshopper.com
Getting Private key
```

You can use the `openssl` command to inspect the content of the certificate, too.

```
# openssl x509 -text -in /etc/apache2/ssl.crt/server.crt#
```

As a final test of our configuration, we start the server and ensure that everything comes up as expected. To do this, we need to create a file called



local.conf in /etc/apache2/ and add the following text to tell Apache to use SSL and where to find its certificate and private key:

```
SSL Engine On#
SSLCertificateFile /etc/apache2/
ssl.crt/server.crt#
SSLCertificateKeyFile /etc/apache2/
ssl.key/server.key#
```

Now add the following line to /etc/apache2/httpd.conf:

```
Include local.conf#
```

Start the Apache server using the command:

```
# /etc/init.d/apache2 startssl#
```

This will set the SSL flag. If you get a 'failed' response from the startssl command, check you've entered everything correctly.

MAKING CONNECTIONS

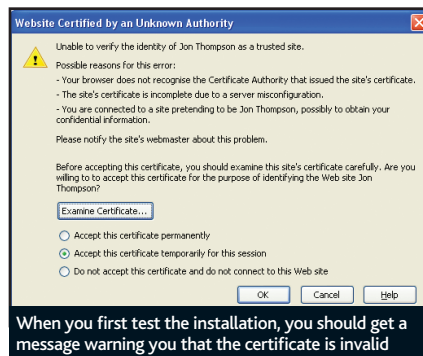
We now test the mod_ssl installation by pointing a web browser at the server. If your server has the IP address 192.168.0.3, you would enter https://192.168.0.3/ in your browser and press return. Note that it is https and not http at the beginning of the address. You should get a pop-up explaining that the server's certificate is invalid and someone may be trying to direct you to a fake site (see the screen above). Press OK to accept this and any other certificate warnings that appear. In your browser, you should see the Apache web page. There should also be a padlock in the browser's display, probably at the bottom-right of the window. In Firefox, the URL bar will turn beige. These are all signs that SSL is running on the server. If your browser displays a message saying you've made a bad request, check the URL to ensure you have used https.

We know the server is up, running and serving SSL-protected content if we use https:// in the URL. It's inconvenient to have to do that, though. It would be better if Apache automatically took any http:// request and redirected the connection to the SSL system. Apache has a useful facility called virtual hosting that we can use to achieve this by automatically redirecting traffic that appears on port 80 to port 443. Virtual hosting is a way of storing many websites on a single server with a single IP address. The domain name is part of the HTTP connection request. When Apache receives the request for a website at a certain domain, it uses the VirtualHost sections in its configuration to determine where to find content for that site. On a fast enough server, it is possible to host a large number of websites in this way.

For testing purposes, we need to add a temporary entry to a file on the client computer – the same one we're using to browse the web. The hosts file lists server names and IP addresses. It overrides normal domain name system (DNS) lookups and is ideal for testing. We must ensure that entering the fake www.cshopper.co.uk URL won't cause the browser to perform a DNS lookup and try to find the site out on the internet. If you use Linux, add this line to /etc/hosts:

```
192.168.0.3 www.cshopper.com#
```

If you use Windows, add exactly the same line to



the host file you'll find in C:\WINDOWS\system32\drivers\etc\hosts.

Substitute your server's IP address for 192.168.0.3 as appropriate. Edit the local.conf file created earlier and overwrite it with the following:

```
SSLCertificateFile /etc/apache2/
ssl.crt/server.crt#
SSLCertificateKeyFile /etc/apache2/
ssl.key/server.key#
NameVirtualHost *:80#
<VirtualHost *:80>#
    ServerName www.cshopper.com#
    SSL Engine off#
    DocumentRoot /srv/www/htdocs#
    <Location />#
        RedirectPermanent / https://
www.cshopper.com/#
    </Location>#
</VirtualHost>#
#
NameVirtualHost *:443#
<VirtualHost *:443>#
    ServerName www.cshopper.com#
    SSL Engine on#
    DocumentRoot /srv/www/htdocs#
</VirtualHost>#
```

Each VirtualHost section has a ServerName definition and configuration commands that apply only to that host, such as whether to turn on SSL. We've defined two versions of the same host: one using port 80, the other using port 443. The asterisks in the virtual host definitions indicate that we can connect to the host via any available IP address on the server.

If we now try to connect to the http://www.cshopper.com address, the first virtual host definition handles requests that arrive for that site on port 80. The location section ensures that requests for any location, from the root directory for the site downwards, immediately redirect to https://www.cshopper.com.

This is where the second virtual host definition takes over. It attempts to establish an SSL connection and then serves the relevant content. The user sees that a connection to a page on www.cshopper.com brings up the secure version of the page.

Apache has lots of configuration options, known as directives. Tasks such as this can be carried out using many different means; for instance, by having Apache rewrite the URL. It's worth studying the online documentation at <http://httpd.apache.org/docs-2.0> to familiarise yourself with what's available.

As a final security measure, let's change the permissions on the all-important, secret files

REVIEW OPERATING SYSTEM

NOVELL SuSE Linux Professional 9.3

RATING ★★★★★

PRICE £47 inc VAT

SUPPLIER www.amazon.co.uk

DETAILS www.novell.com/linux/suse

Novell's SuSE Professional Linux 9.2 was always going to be a hard act to follow. In *Shopper* March 2005 we gave it the thumbs up. But just a few months later, there's another release that promises more. So does this new distribution match the rate of progress set by its predecessor? And might Novell be the first Linux producer to give Microsoft a bloody nose?

This is a big distribution, and it comes complete with both CD and DVD media. The CD set runs to five discs and the two DVDs hold the full 32- and 64-bit systems. Should you wish to build your own setup, there's also the complete source code tree for the operating system.

The documentation is extensive, too, and comes in two volumes. The User Guide is a no-nonsense introduction to what's available. The larger Administration Guide covers the essentials of running the various parts of the distribution, including all the bundled servers.

A full installation takes just under 7GB of disk space, but a minimal desktop system is a more manageable 2GB, plus 500MB for virtual memory. YaST, the graphical system configuration tool, handles the installation, which now asks for your agreement to install Macromedia's Flash Player 7. Although some open source purists might balk at this inclusion, most will be glad they don't have to download it themselves.

Once installed and booted, there are several desktop environments to choose from, including KDE 3.4.0.28 and Gnome 2.10.0-4. KDE will be familiar to Windows desktop users, with its horizontal taskbar and a pop-up launch menu at the bottom left. Gnome, on the other hand, will be more familiar to users of the Apple Macintosh.

The most significant part of this release is the inclusion of several Voice over IP (VoIP) applications. Linphone is a free VoIP program that allows you to talk to other users anywhere in the world without telephone connection charges. You

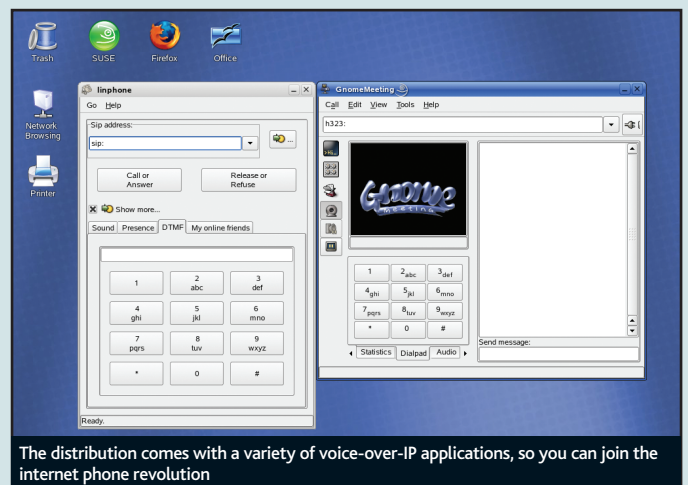
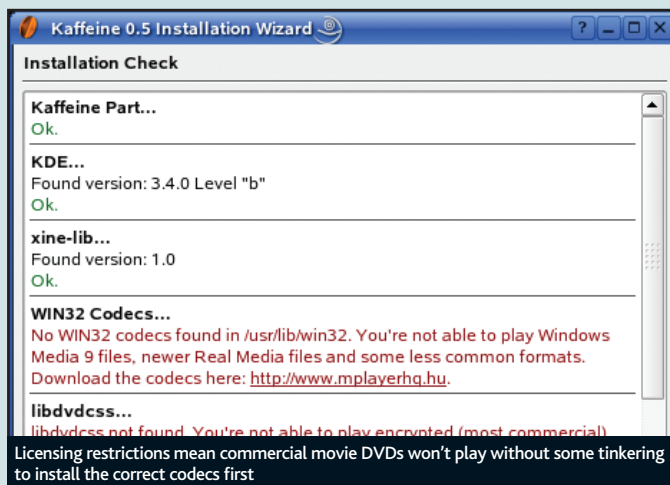
server.key and server.crt. Others must never have access to these:

```
# chmod 400 /etc/apache2/ssl.crt/
server.crt /etc/apache2/ssl.ey/
server.key#
```

Now we have our server running SSL, automatically securing all HTTP traffic, we can perform a speed test to check what the performance is like:

```
# openssl speed#
```

This exercises the mod_ssl installation, putting it through its paces to encrypt various blocks of data



use the internet as a phone network and all you need is a sound card, speakers and microphone to start making calls. Kphone, another similar program, is also included. This allows video calling, as does Gnome Meeting, which also handles teleconferencing.

SuSE 9.3 caters for traditional methods of electronic communication, too. Novell Evolution and KDE's Kontact both handle email, while the excellent Gaim and Kopete allow the user to contact friends and relatives over a wide range of instant messaging systems, including AOL Instant Messenger, ICQ, IRC and Microsoft's MSN.

Audio CDs will play without any problems, but DVDs are another matter. Due to licensing restrictions on proprietary video codecs, the bundled applications won't play most DVDs unless you first find and install the correct software. This is where Microsoft Windows XP scores; it includes, or provides easy access to, the required codecs. SuSE 9.3 has been designed so it is easy to install on the same system as Windows, which means you can freely boot between the two. If switching back into Windows to watch a film is not a problem for you, the lack of codecs won't be an issue.

For bundled games, this distribution wins hands-down. There are over 90 games, from Chess to Asteroids. There are also over 80 traditional UNIX text-based games and pastimes, including an implementation of the original adventure game Colossal Cave.

SuSE 9.3 comes with the excellent Gwenview and F-spot photo indexers, along with the digital photography application Digikam, which Novell

claims is now compatible with most digital cameras. Digikam is used to download photos from the camera to the PC's hard disk. The distribution also includes GIMP 2.2, which can manipulate images to the same standard as expensive Windows packages. In addition, there are two 3D graphics packages: Blender and KPovModeller.

There are several bundled word processors, spreadsheets, databases, presentation packages and address books. This is in addition to the pre-release version of OpenOffice.org 2.0, a mature, largely compatible rival to Microsoft Office that even writes PDF files. You'll need something to view PDF files, of course; luckily, the distribution also carries Adobe Acrobat for Linux.

For those who like to get down and dirty with the operating system, there are several command-line shells. The distribution also carries a full set of system development tools. Everything is bundled, from editors, code management systems, compilers and interpreters to full visual development environments such as KDevelop, Eclipse and Novell's own Mono, which can run Microsoft .NET programs.

This distribution has more toys than Woolworths. Particularly fascinating is a program called KSimus, which enables you to create and run virtual digital electronic circuits and devices in real time. Other intriguing packages include programs to help increase your word power, learn Japanese and even brush up on your Latin. Also included is a preview copy of the new Beagle desktop search tool. Just as a search engine indexes millions of

websites, Beagle's job is to index information on your computer and so put everything at your fingertips. It can find everything from documents, web search histories and instant messaging conversation logs to images and source code.

Xen is another useful addition. This virtual hardware program enables you to run multiple operating systems. Each instance of an operating system believes it is running on its own hardware and this enables several servers to coexist on one physical computer, reducing hardware and management costs. Sadly, Windows is not supported yet.

Linux is now a mature, stable operating system. SuSE Linux Professional 9.3 is a fully featured distribution that wouldn't look out of place next to Windows XP. With so much on offer, there's always the danger of having too much of a good thing. Few people need nine different pocket calculators, even if some are faithful simulations of real life graphing and programmable models.

It's probably not worth upgrading from version 9.2, but if you want to install a brand new Linux system from scratch and you don't want to spend time downloading shed-loads of software before you start working, SuSE Linux Professional 9.3 is extremely good value.

 Clean installation; masses of software; VoIP

 Still missing some codecs for licensing reasons

REQUIREMENTS Pentium 4 processor, 256MB RAM, 2.5GB disk space. **Product code** B0007XQBWQ

using the different algorithms available to it. Some timings seem slow, which is why SSL connections use the minimum asymmetric encryption.

TROUBLESHOOTING

If you encounter problems when using mod_ssl, your first port of call should be the Apache error log in /var/log/apache2. You can see the last few entries by typing:

```
# tail /var/log/apache2/error_log
```

Starting Apache with the start option rather than startssl is a common mistake. Start will cause the server to ignore all SSL configuration information. You can get round this by modifying

the /etc/init.d/apache2 script so that it always starts with the -DSSL option. Remember to make the same changes if you upgrade your installation, however, as this file may be overwritten.

Whenever you decide to make any configuration changes to Apache, you should take a backup of your configuration files first. There are few things more frustrating than not knowing how to return to a previous working state.

If a newly installed Apache refuses all HTTP and HTTPS connections, check your firewall to ensure you're allowing ports 80 and 443 through. Most mainstream distributions turn the firewall on by default and obey the principle that if a port doesn't need to be open, it will block all connection requests it receives. As installing and running

Apache does not automatically open these ports, you should check this first to save yourself a few frustrating hours.

One problem that affects many busy administrators is forgetting to renew the certificate when it expires. If you start receiving worried complaints about certificate warnings, this may be the cause. It's well worth putting a reminder in Korganiser for next year. **CS**

NEXT MONTH

BUILD YOUR OWN HOME MEDIA SERVER
How to turn the Buffalo Linkstation into a streaming media server